



Experience with Automation

ITk DAQ Workshop

May 29, 2019

Aleksandra Dimitrievska, Maurice Garcia-Sciveres,
Timon Heim, Simone Pagan Griso

Lawrence Berkeley National Laboratory

Automation

- Measurements for Analog Front-End RD53A Review
- Low dose rate irradiation of RD53A chip



Measurements for Analog Front-End RD53A Review

- Tuning and measurement of different operation points combined in one scrips
- labRemote (control power supplies, multimeters) and Yarr
<https://gitlab.cern.ch/berkeleylab/labRemote>
<https://gitlab.cern.ch/YARR/YARR>

```
#!/bin/bash
```

```
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/std_digitalscan.json -p -m 1 -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_analogscan.json -p -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_intimeanalogscan.json -p -m 0 -o $3

./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_tune_globalthreshold.json -t $4 -p -m 0 -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_tune_pixelthreshold.json -t $4 -p -m 0 -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_tune_globalpreamp.json -t 7500 6 -p -m 0 -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_retune_pixelthreshold.json -t $4 -p -m 0 -o $3
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_tune_finepixelthreshold.json -p -m 0 -o $3

./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_thresholdscan.json -p -m 0 -o $3
scripts/plotWithRoot_Threshold ${3}/last_scan pdf 1
scripts/plotWithRoot_ThresholdTDAC ${3}/last_scan pdf 1
scripts/plotWithRoot_NoiseMap ${3}/last_scan pdf 1
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_intimethresholdscan.json -p -m 0 -o $3
scripts/plotWithRoot_Threshold ${3}/last_scan pdf 1
scripts/plotWithRoot_ThresholdTDAC ${3}/last_scan pdf 1
scripts/plotWithRoot_NoiseMap ${3}/last_scan pdf 1
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/std_totscan.json -p -m 0 -t 6000 -o $3
scripts/plotWithRoot_ToT ${3}/last_scan pdf 1
./bin/scanConsole -r $1 -c $2 -s configs/scans/rd53a/diff_noisescan.json -p -o $3
```

Measurements for Analog Front-End RD53A Review

- Tuning and measurement of different operation points combined in one scrips
- labRemote (control power supplies, multimeters) and Yarr

```
#!/bin/bash

echo "Measuring current for a specific loaded configuration"
echo "Need to have labRemote, change the path to the repository"

cd ~/labRemote/build

#Analog
AnalogI=`./bin/agilent_measure -c 1 -p /dev/ttyUSB0 -g 3 get-current`
#AnalogI=`./bin/rigol_measure -c 1 -p /dev/usbtmc0 get-current`
#Digital
DigitalI=`./bin/agilent_measure -c 2 -p /dev/ttyUSB0 -g 3 get-current`
#DigitalI=`./bin/rigol_measure -c 2 -p /dev/usbtmc0 get-current`

echo "Analog current is ${AnalogI}"
echo "Digital current is ${DigitalI}"

cd -

exit 0
```


SOFTWARE

- Monitoring and data acquisition code:
<https://gitlab.cern.ch/berkeleylab/slipper-monitoring-sw>
- Combines: Yarr, labRemote (control power supplies, multimeters), mysql

TESTING PROCEDURE

- Keep chip busy all the time (noise scans with global pulse for ring oscillators)
- Perform scans every hour (threshold, tot, MUX, ring oscillators)
- Tuning (1ke, 7 ToT at 10k e) once a day
- Monitor environmental conditions, humidity and temperature, voltage outputs from the chip every minute via Arduino
- Monitor input current of the chip
- The data is stored in database: Arduino, Chip and Log tables.

○ Cronjobs

```
## Slipper crontab content
# More info: man 5 crontab
# m h dom mon dow  command
*/1 * * * * /local/slipper/slipper-monitoring-sw/scripts/cron_ChipLoop.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
1 22 * * * /local/slipper/slipper-monitoring-sw/scripts/cron_cleanup.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
*/1 * * * * /local/slipper/slipper-monitoring-sw/scripts/cron_Arduino_IO.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
45 * * * * /local/slipper/slipper-monitoring-sw/scripts/cron_FullScans.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
20 17 * * * /local/slipper/slipper-monitoring-sw/scripts/cron_ScanFirstConfig.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
10 8 * * * /local/slipper/slipper-monitoring-sw/scripts/cron_ChipTuning.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
50 23 * * * /local/slipper/slipper-monitoring-sw/scripts/cron_BackupDB.sh >> /local/slipper/slipper-data/perm/log/cron.log 2>&1
```

Low dose rate irradiation

○ Lock File

```
while true; do

# request YARR lock, default timeout
requestYarrLock
if [ $? -ne 0 ]; then
#timed out or other general error
python ${PythonScripts}/Log_Service.py ${CommonCONFIG} --source 3 --type 2 \
--message "${progName} failed requesting Yarr lock." --db
exit 1
fi

## Run
cd ${CL_TharPath}/run

#save results qwith a consistent timestamp
TS=`date +%s`
logTS "ChipLoop TS"

#noise scan with global pulse for ring oscillators (either we run noiseGPscan or analogGPscan which also has the global pulse for the ring oscillators
timeout 10m ../bin/scanConsole -r configs/controller/specCfg.json -c configs/connectivity/example_rd53a_setup.json -s configs/scans/rd53a/std_noise0Pscan.js
n -p -m 0 >& ${CL_RollingScanLogs}/last.log

#release lock file
releaseYarrLock
```

- Check if noise scans are running

```
#!/bin/bash
# Check if ChipLoop is running, if not submit new job

## Init

baseDir=`dirname $0`
progName=`basename $0`
source $baseDir/config.sh

#check if it is running
if [ -w /local/slipper/slipper-data/perm/log/ChipLoop_PID.log ]
then
    PID=`cat /local/slipper/slipper-data/perm/log/ChipLoop_PID.log`
    if ps -p $PID > /dev/null
    then
        LogTS "ChipLoop check, PID ${PID}"
        exit
    fi
fi

#submit new job
CurrentDate=`date +%Y-%m-%d`
nohup sh /local/slipper/slipper-monitoring-sw/scripts/ChipLoop.sh >& /local/slipper/slipper-data/perm/log/${CurrentDate}_CL.log &
New_PID=$!
echo ${New_PID} > /local/slipper/slipper-data/perm/log/ChipLoop_PID.log
python ${PythonScripts}/Log_Service.py ${CommonCONFIG} --source 3 --type 5 \
--message "${progName} is submitting new ChipLoop job with PID ${New_PID}." --db
if [ -z "${New_PID}" ]
then
    python ${PythonScripts}/Log_Service.py ${CommonCONFIG} --source 3 --type 6 \
--message "${progName} failed to submit new ChipLoop job." --db
fi
```

○ Quick data analysis

```
#run threshold scan and save results
timeout 15m ../bin/scanConsole -r configs/controller/specCfg.json -c configs/connectivity/example_rd53a_setup.json -s configs/scans/rd53a/std_thresholdscan.json
-p -m 0 >& ${CL_RollingScanLogs}/last.log
folderToSave='ls -ltr data/ | tail -1'
cp -r data/${folderToSave} ${CL_ScansPath}/${FullScan_TS}/
cp ${CL_RollingScanLogs}/last.log ${CL_ScansPath}/${FullScan_TS}/std_thresholdscan.log
timeout 5m ../external/YARR/src/scripts/plotWithRoot_Threshold ${CL_ScansPath}/${FullScan_TS}/${folderToSave} >& rootoutput.log
ThrScan_Mean_FE_1_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 1p'
ThrScan_Sigma_FE_1_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 1p'
ThrScan_Mean_FE_2_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 2p'
ThrScan_Sigma_FE_2_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 2p'
ThrScan_Mean_FE_3_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 3p'
ThrScan_Sigma_FE_3_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 3p'
insertChipData="UPDATE ${CL_ChipTable} SET ThrScan_Mean_FE_1 = ${ThrScan_Mean_FE_1_Value}, ThrScan_Mean_FE_2 = ${ThrScan_Mean_FE_2_Value}, ThrScan_Mean_FE_3 = ${ThrScan_Mean_FE_3_Value}, ThrScan_Sigma_FE_1 = ${ThrScan_Sigma_FE_1_Value}, ThrScan_Sigma_FE_2 = ${ThrScan_Sigma_FE_2_Value}, ThrScan_Sigma_FE_3 = ${ThrScan_Sigma_FE_3_Value}"
insertChipData="${insertChipData} WHERE TS = ${FullScan_TS}"
mysql -h $(host) -u $(LogUser) -p${LogPassword} -e "use ${Database}; ${insertChipData};"
mysql -h $(host) -u $(LogUser) -p${LogPassword} -e "use ${Database}; ${insertChipData};"
cp ${CL_ScansPath}/${FullScan_TS}/${folderToSave} >& rootoutput.log
ThrScan_Noise_Mean_FE_1_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 1p'
ThrScan_Noise_Sigma_FE_1_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 1p'
ThrScan_Noise_Mean_FE_2_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 2p'
ThrScan_Noise_Sigma_FE_2_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 2p'
ThrScan_Noise_Mean_FE_3_Value='less rootoutput.log | grep FE | awk '{print $2}' | sed -n 3p'
ThrScan_Noise_Sigma_FE_3_Value='less rootoutput.log | grep FE | awk '{print $3}' | sed -n 3p'
insertChipData="UPDATE ${CL_ChipTable} SET ThrScan_Noise_Mean_FE_1 = ${ThrScan_Noise_Mean_FE_1_Value}, ThrScan_Noise_Mean_FE_2 = ${ThrScan_Noise_Mean_FE_2_Value}, ThrScan_Noise_Mean_FE_3 = ${ThrScan_Noise_Mean_FE_3_Value}, ThrScan_Noise_Sigma_FE_1 = ${ThrScan_Noise_Sigma_FE_1_Value}, ThrScan_Noise_Sigma_FE_2 = ${ThrScan_Noise_Sigma_FE_2_Value}, ThrScan_Noise_Sigma_FE_3 = ${ThrScan_Noise_Sigma_FE_3_Value}"
insertChipData="${insertChipData} WHERE TS = ${FullScan_TS}"
mysql -h $(host) -u $(LogUser) -p${LogPassword} -e "use ${Database}; ${insertChipData};"
cp ${CL_ScansPath}/${FullScan_TS}/${folderToSave}/${SSL_ChipName_SCC_A}_ThresholdMap_PLOT.png $(WebHome)/Figures/ThMap_SCC_A.png
cp ${CL_ScansPath}/${FullScan_TS}/${folderToSave}/${SSL_ChipName_SCC_A}_ThresholdMap_STACK.png $(WebHome)/Figures/ThDist_SCC_A.png
cp ${CL_ScansPath}/${FullScan_TS}/${folderToSave}/${SSL_ChipName_SCC_A}_NoiseMap_PLOT.png $(WebHome)/Figures/NoiseMap_SCC_A.png
cp ${CL_ScansPath}/${FullScan_TS}/${folderToSave}/${SSL_ChipName_SCC_A}_NoiseMap_STACK.png $(WebHome)/Figures/NoiseDist_SCC_A.png
```

Low dose rate irradiation

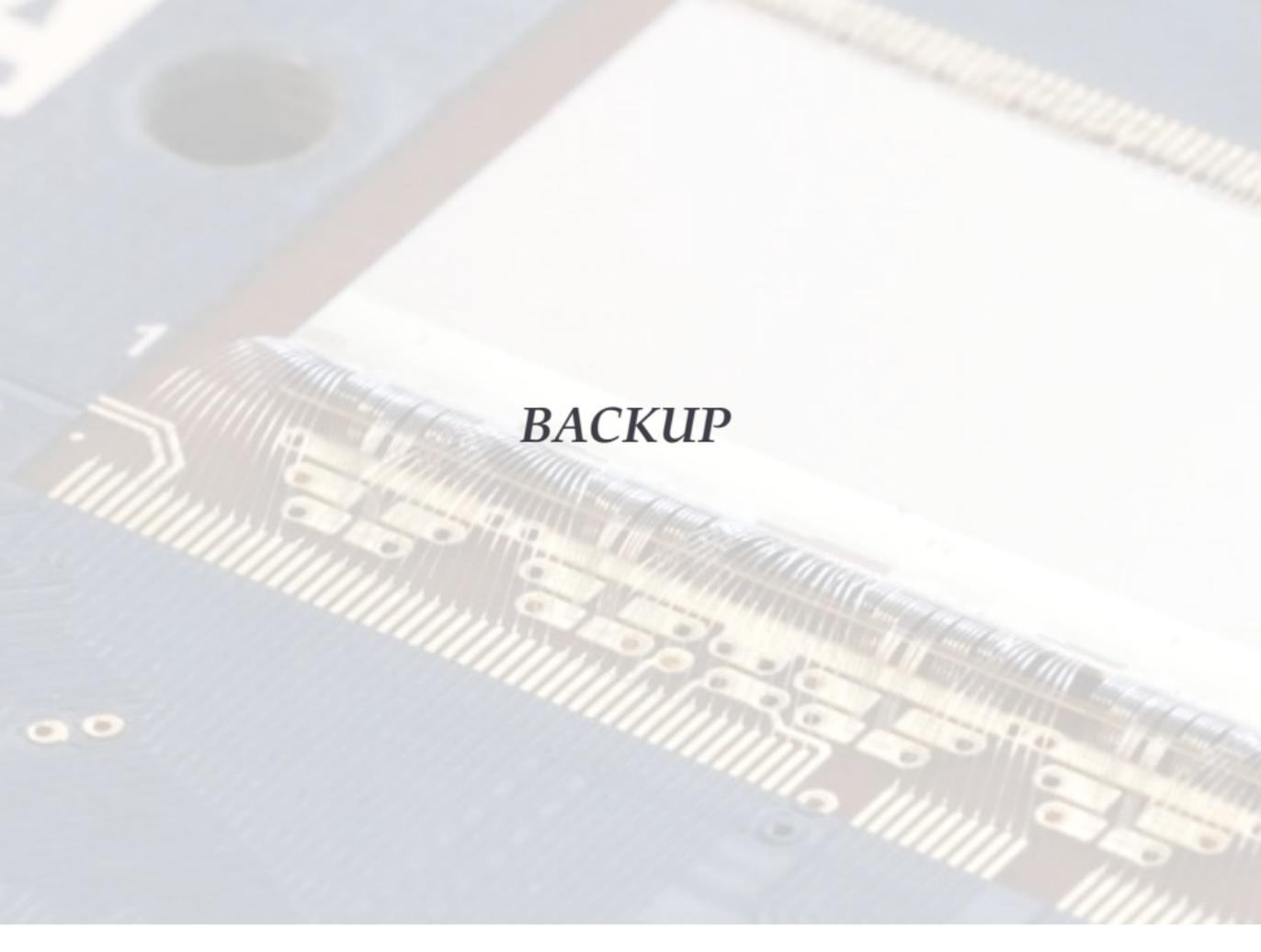
○ Check if tuning failed

```
#check if the value of the Lin FE threshold and use default pretuned config file if the tuning failed
LinTH_Value=`grep LinVth configs/rd53a_test.json | awk '{print $2}' | cut -d',' -f 1`
if [ ${LinTH_Value} -eq 0 ]; then
    cp configs/rd53a_tunedAllCold.json configs/rd53a_test.json
    python ${PythonScripts}/Log_Service.py ${CommonCONFIG} --source 3 --type 4 \
        --message "${progName} failed tuning of the LinFE." --db
fi
```

○ Check if power supply is on

```
#check PS status and take actions if necessary
requestUSBLOCK
V_1=`./bin/rigol_ctrl --port ${SSL_PS_1} --channel 1 get-voltage`
V_1_on=`echo "${V_1} > 0.1" | bc`
V_2=`./bin/rigol_ctrl --port ${SSL_PS_1} --channel 2 get-voltage`
V_2_on=`echo "${V_2} > 0.1" | bc`
if [[ "${V_1_on}" -eq 0 ]] && [[ "${V_2_on}" -eq 0 ]]; then
    #PS went off. check last time it happened
    lastPSActionTS=`tail -1 .lastPSAction`
    [[ -z "${lastPSActionTS}" ]] && lastPSActionTS=0
    lastPSActionInterval=$(( TS - lastPSActionTS ))

    #if more than 1hr ago, try to power it on again
    #and save now as last time this was powered on
    if [[ ${lastPSActionInterval} -ge 1800 ]]; then
        ./bin/rigol_ctrl --port ${SSL_PS_1} --channel 2 power-on 1.85 2.0
        sleep 1
        ./bin/rigol_ctrl --port ${SSL_PS_1} --channel 1 power-on 1.85 2.0
        echo "${TS}" >> .lastPSAction
        python ${PythonScripts}/Log_Service.py ${CommonCONFIG} --source 3 --type 1 --timestamp ${TS} \
            --message "Powered ON PS" --db
    else
        echo "${TS}" >> .PSOffNoAction
    fi
fi
```

A close-up photograph of a blue printed circuit board (PCB) with a gold-plated connector strip. The strip features numerous gold pins and pads. The word "BACKUP" is printed in a bold, black, serif font across the center of the image. The background is a light, neutral color.

BACKUP

Low dose rate irradiation

- Low dose rate irradiation of RD53A chip with Kr-85

